CLAIMS

What is claimed is:

1. A method for differentiating congestion-related packet loss versus random packet loss in a wireless data connection, comprising:

monitoring changes in the length of a transmission queue in a wireless data connection;

designating packet loss as being due to congestion if said packet loss is preceded by an increase in the queue length; and

designating packet loss as random loss if said packet loss is not preceded by an increase in the queue length.

2. A method as recited in claim 1, further comprising:

applying a collision avoidance algorithm if packet loss is designated as due to congestion.

3. A method as recited in claim 2, wherein said collision avoidance algorithm comprises reducing the sender's transmission window by one-half.

4. A method as recited in claim 1, further comprising:

monitoring changes in the length of said queue over an interval substantially equal to the amount of time it takes to transmit a window of data packets and receive acknowledgements corresponding to all data packets transmitted in the window.

5. A method as recited in claim 4, further comprising:

initializing a state count to zero;

transitioning from the state count zero to state count one if the length of said queue increases during the next interval;

transitioning from the state count one to state count zero if the length of said queue decreases or remains steady during the next subsequent interval;

transitioning from state count one to state count two if the length of said queue increases during the next subsequent interval; and

designating packet loss as due to congestion if state count two is reached.

6. A method as recited in claim 5, further comprising:

applying a collision avoidance algorithm if packet loss is designated as due to congestion.

7. A method as recited in claim 6, wherein said collision avoidance algorithm comprises reducing the sender's transmission window by one-half.

8. A method as recited in claim 1, further comprising determining whether congestion is developing in the forward or reverse path of the connection.

9. A method as recited in claim 8, further comprising isolating forward throughput from congestion on the reverse path.

10. A method as recited in claim 9, wherein congestion is determined by calculating the relative delay that one packet experiences with respect to another as it traverses the connection.

11. A method as recited in claim 10, wherein said relative delay is used to estimate the number of packets residing the in the queue.

12. A method as recited in claim 11, further comprising keeping the number of packets in the queue at a minimum level by adjusting a congestion window.

13. A method as recited in claim 12, further comprising:

reducing the congestion window if the queue length increases; and

increasing the congestion window if the queue length decreases.

14. A TCP-based congestion management protocol for a wireless data connection, comprising:

monitoring changes in the length of a transmission queue in a data connection;

designating packet loss as being due to congestion if said packet loss is preceded by at least two consecutive intervals of increasing queue length; and

designating packet loss as random loss if said packet loss is not preceded by at least two consecutive intervals of increasing queue length.

15. A method as recited in claim 14, further comprising:

applying a collision avoidance algorithm if packet loss is designated as due to congestion.

16. A method as recited in claim 15, wherein said collision avoidance algorithm comprises reducing the sender's transmission window by one-half.

17. A protocol as recited in claim 16, wherein each said interval comprises the amount of time it takes to transmit a window of data packets and receive acknowledgements corresponding to all data packets transmitted in the window.

18. A protocol as recited in claim 17, further comprising:

initializing a state count to zero;

transitioning from the state count zero to state count one if the length of said queue increases during the next interval;

transitioning from the state count one to state count zero if the length of said queue decreases or remains steady during the next subsequent interval;

transitioning from state count one to state count two if the length of said queue increases during the next subsequent interval; and

designating packet loss as due to congestion if state count two is reached.

19. A protocol as recited in claim 18, further comprising:

applying a collision avoidance algorithm if packet loss is designated as due to congestion.

20. A protocol as recited in claim 19, wherein said collision avoidance algorithm comprises reducing the sender's transmission window by one-half.

21. A protocol as recited in claim 14, further comprising determining whether congestion is developing in the forward or reverse path of the connection.

22. A protocol as recited in claim 21, further comprising isolating forward throughput from congestion on the reverse path.

23. A protocol as recited in claim 22, wherein congestion is determined by calculating the relative delay that one packet experiences with respect to another as it traverses the connection.

24. A protocol as recited in claim 23, wherein said relative delay is used to estimate the number of packets residing the in the queue.

25. A protocol as recited in claim 24, further comprising keeping the number of packets in the queue at a minimum level by adjusting a congestion window.

26. A protocol as recited in claim 25, further comprising:

reducing the congestion window if the queue length increases; and

increasing the congestion window if the queue length decreases.

27. A method for differentiating congestion-related packet loss versus random packet loss in a wireless data connection, comprising:

monitoring changes in the length of a transmission queue in a wireless data connection over an interval substantially equal to the amount of time it takes to transmit a window of data packets and receive acknowledgements corresponding to all data packets transmitted in the window;

designating packet loss as being due to congestion if said packet loss is preceded by an increase in the queue length; and

designating packet loss as random loss if said packet loss is not preceded by an increase in the queue length.

28. A method as recited in claim 27, further comprising:

applying a collision avoidance algorithm if packet loss is designated as due to congestion.

29. A method as recited in claim 28, wherein said collision avoidance algorithm comprises reducing the sender's transmission window by one-half.

30. A method as recited in claim 27, further comprising:

initializing a state count to zero;

transitioning from the state count zero to state count one if the length of said queue increases during the next interval;

transitioning from the state count one to state count zero if the length of said queue decreases or remains steady during the next subsequent interval;

transitioning from state count one to state count two if the length of said queue increases during the next subsequent interval; and

designating packet loss as due to congestion if state count two is reached.

31. A method as recited in claim 30, further comprising:

applying a collision avoidance algorithm if packet loss is designated as due to congestion.

32. A method as recited in claim 31, wherein said collision avoidance algorithm comprises reducing the sender's transmission window by one-half.

33. A method as recited in claim 27, further comprising determining whether congestion is developing in the forward or reverse path of the connection.

34. A method as recited in claim 33, further comprising isolating forward throughput from congestion on the reverse path.

35. A method as recited in claim 34, wherein congestion is determined by calculating the relative delay that one packet experiences with respect to another as it traverses the connection.

36. A method as recited in claim 35, wherein said relative delay is used to estimate the number of packets residing the in the queue.

37. A method as recited in claim 36, further comprising keeping the number of packets in the queue at a minimum level by adjusting a congestion window.

38. A method as recited in claim 37, further comprising:

reducing the congestion window if the queue length increases; and

increasing the congestion window if the queue length decreases.

39. A TCP-based congestion management protocol for a wireless data connection, comprising:

monitoring changes in the length of a transmission queue in a data connection over an interval substantially equal to the amount of time it takes to transmit a window of

data packets and receive acknowledgements corresponding to all data packets transmitted in the window;

designating packet loss as being due to congestion if said packet loss is preceded by at least two consecutive intervals of increasing queue length; and

designating packet loss as random loss if said packet loss is not preceded by at least two consecutive intervals of increasing queue length.

40. A method as recited in claim 39, further comprising:

applying a collision avoidance algorithm if packet loss is designated as due to congestion.

41. A method as recited in claim 40, wherein said collision avoidance algorithm comprises reducing the sender's transmission window by one-half.

42. A protocol as recited in claim 41, further comprising:

initializing a state count to zero;

transitioning from the state count zero to state count one if the length of said queue increases during the next interval;

transitioning from the state count one to state count zero if the length of said queue decreases or remains steady during the next subsequent interval;

transitioning from state count one to state count two if the length of said queue increases during the next subsequent interval; and

designating packet loss as due to congestion if state count two is reached.

43. A protocol as recited in claim 42, further comprising:

applying a collision avoidance algorithm if packet loss is designated as due to congestion.

44. A protocol as recited in claim 43, wherein said collision avoidance algorithm comprises reducing the sender's transmission window by one-half.

45. A protocol as recited in claim 39, further comprising determining whether congestion is developing in the forward or reverse path of the connection.

46. A protocol as recited in claim 45, further comprising isolating forward throughput from congestion on the reverse path.

47. A protocol as recited in claim 46, wherein congestion is determined by calculating the relative delay that one packet experiences with respect to another as it traverses the connection.

48. A protocol as recited in claim 47, wherein said relative delay is used to estimate the number of packets residing the in the queue.

49. A protocol as recited in claim 48, further comprising keeping the number of packets in the queue at a minimum level by adjusting a congestion window.

50. A protocol as recited in claim 49, further comprising:

reducing the congestion window if the queue length increases; and

increasing the congestion window if the queue length decreases.

51. A method for improving TCP performance over a wireless connection, comprising:

detecting the initial stages of congestion in the connection, and identifying the direction of the congestion;

determining whether congestion is developing in the forward or reverse path of the connection;

isolating the forward throughput from events such as congestion that may occur on the reverse path;

determining congestion by calculating the relative delay that one packet experiences with respect to another as it traverses the network;

using said relative delay to estimate the number of packets residing in a bottleneck queue;

keeping the number of packets in the bottleneck queue at a minimum level by adjusting the TCP source's congestion window;

reducing the congestion window if the bottleneck queue length increases; and

increasing the congestion window when the source detects additional bandwidth availability in the connection.